

EV355227798

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Shared Network Path Contention Reduction**

Inventors:

**Sukesh Pai**

**Wallace Colyer**

**Vivek Thukral**

**Leslie H. Thaler**

ATTORNEY'S DOCKET NO. MS1-1487US

## **TECHNICAL FIELD**

This invention relates to bandwidth management and, in particular, to managing bandwidth to reduce contention in a shared network path.

## **BACKGROUND**

With technological advances of recent years, television viewing is becoming more of an interactive experience. For example, rather than just broadcasting a television signal to be received by an antenna, cable television systems have been developed to broadcast television signals as well as other information, such as an up to date electronic program guide (EPG), to television set-top boxes that receive and process the data. High-end set-top boxes may be configured to support other applications as well, such as an Internet browser application. Other interactive applications may also be enabled over a cable television network, such as a user viewing log collection application, an instant messaging application, an email application, and so on.

Cable television systems (herein referred to as a headend) typically broadcast television programs over what is referred to as an in-band network, and broadcast EPG data and other such data over what is referred to as an out-of-band network. An in-band network may typically include hundreds of channels, each having a large bandwidth (e.g., capable of carrying 38 Mbps). The in-band network is used for one-way communication of data from the headend to the set-top boxes.

On the other hand, the out-of-band network typically consists of only one channel, and supports two-way communication, but with very limited bandwidth that must be shared by multiple set-top boxes on a particular network.

1 Furthermore, the bandwidth available for transmitting data to the set-top boxes  
2 may be greater than the bandwidth available for receiving data from the set-top  
3 boxes. For example, a typical out-of-band channel may support the transfer of  
4 2Mbps of data from the headend to the set-top boxes (i.e., downstream) and  
5 256Kbps of data from the set-top boxes to the headend (i.e., upstream).

6 Furthermore, most cable television systems are configured to use a data  
7 transfer protocol in which it is common for data packets to collide and be lost,  
8 with no notification to the sender that the data was lost.

9 With an increased desire for interactive data applications to be available to  
10 cable television viewers over the cable television broadcast system, a need exists  
11 for reliable two-way data communication between a headend and multiple set-top  
12 boxes over the out-of-band network.

## 13 14 **SUMMARY**

15 Methods and systems for reducing network contention in a shared network  
16 path environment are described. In the described exemplary environment,  
17 upstream network path bandwidth is significantly smaller than corresponding  
18 downstream network path bandwidth. Also, multiple upstream network paths may  
19 correspond to a single downstream network path, and multiple client devices may  
20 share the same upstream path.

21 In one implementation, data collisions along an upstream network path are  
22 reduced by limiting the number of client devices that share a particular upstream  
23 network path to which pending requests may be outstanding. In an alternate  
24 implementation, a request is not sent to a client device until it is verified that a  
25 sufficient amount of downstream bandwidth is available based on the request size,

1 and it is verified that a sufficient amount of upstream bandwidth is available based  
2 on an anticipated size of a response to the request. An exemplary system may also  
3 be configured to further limit the amount of upstream and/or downstream  
4 bandwidth that is available to requests received from one or more particular  
5 applications.

## 6 7 **BRIEF DESCRIPTION OF THE DRAWINGS**

8 The same numbers are used throughout the drawings to reference like  
9 features and components.

10 Figure 1 is a block diagram illustrating an exemplary cable television  
11 network system.

12 Figure 2 is a block diagram illustrating selected components of, and a data  
13 flow path through, an exemplary cable television system headend.

14 Figure 3 is a block diagram illustrating exemplary downstream paths of an  
15 out-of-band network in the cable television system of Figure 1.

16 Figure 4 is a block diagram illustrating exemplary upstream paths of an out-  
17 of-band network in the cable television system of Figure 1.

18 Figure 5 is a block diagram illustrating data flow through an exemplary  
19 interactive application gateway component of the headend illustrated in Figure 2.

20 Figure 6 is a block diagram illustrating the transfer of requests from a first  
21 upstream path request queue to a corresponding upstream path dispatch queue.

22 Figure 7 is a block diagram illustrating the transfer of requests from a  
23 second upstream path request queue to a corresponding upstream path dispatch  
24 queue.  
25

1        Figure 8 is a block diagram illustrating processing performed to determine  
2 whether sufficient bandwidth is available to transfer a request from an upstream  
3 path request queue to a corresponding upstream path dispatch queue.

4        Figure 9 is a block diagram illustrating data flow through an exemplary  
5 interactive application gateway that includes application request channels.

6        Figure 10 is a block diagram illustrating processing performed to determine  
7 whether sufficient bandwidth is available to transfer requests from upstream path  
8 request queues to corresponding upstream path dispatch queues.

9        Figure 11 is a block diagram illustrating negative effects of repeated request  
10 failures.

11       Figure 12 is a block diagram illustrating an upstream path dispatch queue  
12 over time in a system configured to perform failure handling.

13       Figure 13 is a flow diagram of an exemplary method for reducing network  
14 contention when scheduling transmission of requests.

15       Figure 14 is a flow diagram of an exemplary method for reducing network  
16 contention based on bandwidth when scheduling transmission of requests.

17       Figure 15 is a flow diagram of an exemplary method for gathering data to  
18 be used to reduce network contention when scheduling transmission of requests.

19       Figure 16 is a flow diagram of an exemplary method for handling request  
20 failures.

## 21 22 **DETAILED DESCRIPTION**

23       The following discussion is directed to methods and systems for enabling  
24 reliable two-way communication over an out-of-band network in a cable television  
25 system. Although described in the context of a cable television system, the

1 methods and systems described herein may be implemented to enable reliable two-  
2 way communication over any network in which the downstream bandwidth differs  
3 from the upstream bandwidth.

#### 4 **Exemplary Environment**

5 Figure 1 illustrates an exemplary environment 100 in which reliable two-  
6 way communication over a shared network may be implemented. Exemplary  
7 environment 100 is illustrated as a cable television system that facilitates  
8 distribution of content and program data to multiple viewers. The  
9 environment 100 includes one or more program data providers 102, one or more  
10 content providers 104, a headend 106, and multiple client devices 108(1), 108(2),  
11 ..., 108(N) coupled to the headend 106 via a broadcast network 110.

12 Content providers 104 provide media content such as movies, television  
13 programs, commercials, music, and similar audio and/or video content to  
14 headend 106. Program data providers 102 provide additional data to headend 106,  
15 such as electronic program guide (EPG) data, advertisements, user interface data,  
16 and so on. EPG data that may be provided typically includes some combination of  
17 program identifiers, program titles, ratings, characters, descriptions, actor names,  
18 station identifiers, channel identifiers, schedule information, and so on.

19 Headend 106 processes the program data received from program data  
20 provider 102 and the content received from content provider 104 prior to  
21 transmitting the program data and content across broadcast network 110.  
22 Headend 106 broadcasts signals, such as cable television signals, across broadcast  
23 network 110.

24 Broadcast network 110 is representative of both an in-band network with  
25 substantial bandwidth, which is used to broadcast media content, and an out-of-

1 band network with very limited bandwidth, which is used to broadcast EPG data.  
2 In the described exemplary implementation, the out-of-band network may also be  
3 used to facilitate two-way communication between an application stored at the  
4 headend 106 and one or more of the client devices 108.

5 Client devices 108 can be implemented in any number of ways. For  
6 example, client devices 108(1) and 108(2) are examples of television set-top boxes  
7 coupled to receive broadcast content from broadcast network 110 and provide the  
8 received content to associated televisions 112(1) and 112(2), respectively. Client  
9 device 108(N) is an example of a combination television 114 and integrated set-  
10 top box 116. In this example, the various components and functionality of the set-  
11 top box are incorporated into the television, rather than using two separate devices.  
12 The set-top box incorporated into the television receives broadcast signals via  
13 broadcast network 110.

#### 14 **Exemplary Cable System Headend**

15 Figure 2 illustrates selected components of an exemplary cable system  
16 headend 106. Headend 106 includes a carousel file system 202, an out-of-band  
17 modulator 204, one or more applications 206(1), 206(2), ..., 206(X), an interactive  
18 application gateway (IAG) 208, a network controller 210, and a return path  
19 demodulator 212. Carousel file system 202 is typically used to store data (e.g.,  
20 EPG data) that can be broadcast over the out-of band network 203 to the multiple  
21 set-top boxes 108. Out of band modulator 204 generates formatted data packets  
22 from data that is received from the carousel file system 202, the network  
23 controller 210, and/or other devices, and outputs the formatted data packets at a  
24 constant bit rate over the out-of-band network 203. Network controller 210 and  
25 IAG 208 work together to control two-way communication over the out-of-band

1 network 203 between one or more applications 206 and one or more set-top  
2 boxes 108. Return path demodulator 212 receives data transmitted from set-top  
3 boxes 108 over out-of-band network 203. Applications 206 may include any type  
4 of interactive applications that require two-way communication. Although not  
5 shown, applications 206 may reside on one or more different application servers  
6 associated with headend 106.

### 7 **Exemplary Communication Paths**

8 Figures 3 and 4 illustrate exemplary downstream and upstream paths over  
9 an out-of-band network between a cable system headend 106 and multiple set-top  
10 boxes (e.g., client devices 108 as illustrated in Figures 1 and 2). In the described  
11 implementation, the broadcast network 110 includes a plurality of distribution  
12 hubs, each having one or more modulators and one or more demodulators.

13 Figure 3 illustrates downstream paths of out-of-band network 203. Each  
14 downstream path of the out-of-band network is defined as a path from the network  
15 controller 210 to a modulator 204 in a distribution hub 304. In the example  
16 illustrated in Figure 3, data (e.g., EPG data or data associated with an interactive  
17 application) is transferred from the headend 106 to set-top box 312 (and any other  
18 set-top boxes associated with modulator 204(1)) over a first downstream path 320.  
19 Similarly, data is transferred from the headend 106 to set-top boxes associated  
20 with modulator 204(2) through distribution hub 304(2) (i.e., set-top boxes 314(1),  
21 314(2), 316(1), 316(2), 316(3)) over a second downstream path 322. Data is  
22 transferred from the headend 106 to set-top box 318 (and any other set-top boxes  
23 associated with modulator 204(3) through distribution hub 304(N) over a third  
24 downstream path 324.  
25



Figure 4 illustrates upstream paths of out-of-band network 203. Each upstream path is defined as path that originates at a demodulator 212. In the example illustrated in Figure 4, data (e.g., a response to a request associated with an interactive application) is transmitted from set-top box 312 to the headend 106 over a first upstream path 402 through demodulator 212(1) of distribution hub 304(1). Similarly, data can be transferred from either set-top box 314(1) or 314(2) to the headend 106 over upstream path 406 through demodulator 212(3). Set-top boxes 316(1), 316(2), and 316(3) communicate with headend 106 over upstream path 408 through demodulator 212(4), and set-top box 318 communicates with headend 106 over upstream path 410 through demodulator 212(5). Upstream paths 404 and 412 enable data transmission from set top boxes (not shown) associated with demodulators 212(2) and 212(6), respectively.

Therefore, according to the downstream and upstream paths shown in Figures 3 and 4, it is clear that set-top boxes that receive data from the headend over the same downstream path may transmit data to the headend over different upstream paths. For example, as shown in Figures 3 and 4, set-top boxes 314(1), 314(2), 316(1), 316(2), and 316(3) all receive data over the same downstream path 322, but set-top boxes 314(1) and 314(2) transmit data back over the upstream path 406, while set-top boxes 316(1), 316(2), and 316(3) transmit data back over the upstream path 408.

Two set-top boxes contend for a particular upstream path if there is a chance that a data packet sent by one of the set-top boxes can be lost due to a collision with a data packet sent by the other set-top box. In the described implementation, there is typically no contention for downstream paths because of

1 an Ethernet link between network controller 210 and each modulator 204. The  
2 Ethernet link has carrier sense with collision detection to prevent data collisions.  
3 Data may be lost over the downstream path, however, if a modulator 204 is sent  
4 data at a faster rate than it can handle. Configuration of a downstream bandwidth  
5 limitation associated with a particular modulator 204 is used to prevent such an  
6 occurrence. Contention does exist, however, for the upstream paths because there  
7 is nothing in place to prevent two set-top boxes sharing the same upstream path  
8 from transmitting data at the same time. Interactive application gateway 208 is  
9 implemented at headend 106 to reduce upstream path contention by managing the  
10 transmission of requests from applications 206 to the set-top boxes.

#### 11 **Exemplary Interactive Application Gateway**

12 Figure 5 illustrates selected components of an exemplary interactive  
13 application gateway 208 within headend 106. Interactive application gateway 208  
14 includes application request queue 502 and bandwidth management system 504.  
15 Application request queue 502 receives, from one or more applications 206,  
16 requests that are to be transmitted to one or more set-top boxes 108 over the out-  
17 of-band network 203. Bandwidth management system 504 includes upstream path  
18 (USP) request queues 506(1), 506(2), 506(3), ..., and 506(Y). Bandwidth  
19 management system 504 further includes corresponding upstream path (USP)  
20 dispatch queues 508(1), 508(2), 508(3), ..., and 508(Y). Each USP request  
21 queue 506 (and each corresponding USP dispatch queue 508) corresponds to an  
22 upstream path of the out-of-band network. Requests from applications 206 are  
23 transferred from application request queue 502 to bandwidth management  
24 system 504 where the requests are distributed to the appropriate USP request  
25 queues 506 based on which set-top box each request is directed to. For example,

1 if USP request queue 506(1) corresponds to upstream path 406 (shown in  
2 Figure 4), then any requests directed to set-top boxes 314(1) or 314(2) are placed  
3 in USP request queue 506(1). Similarly, if USP request queue 506(2) corresponds  
4 to upstream path 408, then any requests directed to set-top boxes 316(1), 316(2),  
5 or 316(3) are placed in USP request queue 506(2).

6 Bandwidth management system 504 applies a scheduling algorithm at  
7 regular intervals to transfer requests from USP request queues 506 to the  
8 corresponding USP dispatch queues 508. The requests are then transferred from  
9 the USP dispatch queues 508 to network controller 210, and then on to out-of band  
10 modulator 204 for transmission across out-of-band network 203 to the specified  
11 set-top boxes.

#### 12 **Exemplary Scheduling Algorithm**

13 As described above, the downstream bandwidth leading to a particular set-  
14 top box may differ from the upstream bandwidth leading from the particular set-  
15 top box. Furthermore, multiple upstream paths may be associated with a single  
16 downstream path. For example, as illustrated in Figures 3 and 4 and described  
17 above, upstream paths 406 and 408 are used to transfer data from set-top  
18 boxes 214(1), 214(2), 216(1), 216(2), and 216(3), while the same set-top boxes all  
19 receive data over downstream path 322.

20 In the described implementation, requests are scheduled at regular intervals,  
21 the length of which, are configurable. The scheduling may be performed by  
22 default, for example, every second. In the described exemplary implementation,  
23 scheduling transmission of requests to multiple set-top boxes is based on several  
24 factors including, the bandwidth of the downstream paths associated with the  
25 requests, the bandwidth of the upstream paths that will be associated with

1 responses to the requests, the size of each request, the expected size of each  
2 requested response, and a time interval during which the requests are to be  
3 scheduled for transmission. Interactive application gateway 208 maintains the  
4 following set of configurable parameters that are used in scheduling the  
5 transmission of application requests:

- 6 • upstream bandwidth, expressed in Kbps (bits per millisecond), for each  
7 upstream path (denoted by **ubw<sub>pathID</sub>**)
- 8 • downstream bandwidth, expressed in Kbps (bits per millisecond), for  
9 each downstream path (denoted by **dbw<sub>pathID</sub>**)
- 10 • time interval for scheduling, expressed in milliseconds (denoted by  
11 **period**)
- 12 • average request data size, expressed in bits (denoted by **RequestSize**)
- 13 • average response data size, expressed in bits (denoted by **ResponseSize**)

14  
15 At the beginning of a scheduling interval, the total number of bits available  
16 on a particular upstream path during the scheduling interval is given by:

$$17 \text{ USPBits}_{\text{pathID}} = \text{ubw}_{\text{pathID}} * \text{period bits.}$$

18  
19 Similarly, the total number of bits available on a particular downstream  
20 path during the same scheduling interval is given by:

$$21 \text{ DSPBits}_{\text{pathID}} = \text{dbw}_{\text{pathID}} * \text{period bits.}$$

22  
23 A thread associated with the bandwidth management system 504 wakes up  
24 every period milliseconds, and transfers requests from the USP request queues 506  
25 to the USP dispatch queues 508. For a given USP request queue, the next request

1 in the queue can be scheduled (i.e., moved to the corresponding USP dispatch  
2 queue) if the upstream path associated with the USP dispatch queue has enough  
3 bandwidth available to support the average response size. As each request is  
4 scheduled, the number of bits available during the scheduling interval on the  
5 upstream and downstream paths are adjusted according to the following equations:

$$6 \quad \text{Remaining USPBits}_{\text{pathID}} = \text{USPBits}_{\text{pathID}} - \text{ResponseSize}$$

$$7 \quad \text{Remaining DSPBits}_{\text{pathID}} = \text{DSPBits}_{\text{pathID}} - \text{RequestSize}$$

8  
9 When a request is added to a particular USP dispatch queue 508, a  
10 dispatcher thread associated with that particular upstream path picks up the request  
11 and initiates communication with the set-top box. When communication is  
12 initiated, the dispatcher thread sends the request data to the set-top box through the  
13 network controller 210 and out-of-band modulator 204.

14 In an exemplary implementation, the time it takes to receive a response to a  
15 request is gathered as requests are received. This data is then used to calculate  
16 running averages of the amount of data exchanged (i.e., RequestSize and  
17 ResponseSize) and the current speed of processing requests. In the described  
18 exemplary implementation, data is gathered and calculated independently for each  
19 application so that, for example, each application has an associated average  
20 request size and average response size. In this way, the IAG 208 can dynamically  
21 adjust the load it puts on the network. For example, if there is a sudden spike in  
22 network activity caused by an external application, the available bandwidth may  
23 be reduced, and the time it takes to process a request may go up. Based on that  
24 information, the IAG 208 reduces the number of requests that are dispatched  
25

1 during a particular scheduling interval. As more bandwidth becomes available, the  
2 number of requests dispatched can go back up.

3       Figures 6-8 illustrate processing that may occur to schedule for dispatch,  
4 requests that are to be sent to four set-top boxes that share a downstream path, but  
5 that are distributed across four different upstream paths. Figure 6 illustrates  
6 transferring of requests from a first USP request queue 506(A) to a corresponding  
7 USP dispatch queue 508(A). Figure 6 is divided into three horizontal sections, as  
8 indicated by the circled numbers, 1-3. At the beginning of the scheduling interval,  
9 USP request queue 506(A) contains nine requests (i.e., A1, A2, A3, A4, A5, A6,  
10 A7, A8, and A9). The available upstream bandwidth is represented by box 602  
11 while the available downstream bandwidth is represented by box 604. Bandwidth  
12 management system 504 determines whether or not the first queued request A1  
13 can be handled by the available downstream bandwidth 604. (This is represented  
14 in Figure 6 by the dashed box A1 within the downstream bandwidth box 604.)  
15 Bandwidth management system 504 also determines whether or not an expected  
16 size of a response to request A1 can be handled by the available upstream  
17 bandwidth 602. (This is represented in Figure 6 by the dashed box A1 Rsp within  
18 the upstream bandwidth box 602.) If the request can be handled by the  
19 downstream bandwidth and the anticipated response to the request can be handled  
20 by the upstream bandwidth, then the request is transferred from the USP request  
21 queue 506(A) to the corresponding USP dispatch queue 508(A).

22       If the first queued request A1 is successfully transferred to the dispatch  
23 queue 508(A), then a similar process is performed with respect to the next queued  
24 request A2. This is illustrated in the second horizontal section of Figure 6,  
25 denoted by the circled 2. In the illustrated example, the available downstream

1 bandwidth 604 can accommodate request A2 and the available upstream  
2 bandwidth 602 can accommodate the anticipated response to request A2.  
3 Accordingly, request A2 is moved from USP request queue 506(A) to the  
4 corresponding USP dispatch queue 508(A). Although request A1 is illustrated as  
5 still being in dispatch queue 508(A), it is likely that request A1 may be dispatched  
6 before request A2 is added to the dispatch queue.

7 The third horizontal section of Figure 6, denoted by the circled 3, illustrates  
8 a similar process performed with respect to the next queued request A3. In the  
9 illustrated example, the available downstream bandwidth 604 can accommodate  
10 the request A3, but the available upstream bandwidth 602 cannot accommodate the  
11 anticipated response to request A3. Accordingly, queued request A3 is not  
12 transferred to USP dispatch queue 508(A), but remains in USP request  
13 queue 506(A). Because the available upstream bandwidth cannot accommodate  
14 the anticipated response to the next queued request, bandwidth management  
15 system 504 begins scheduling requests associated with the next upstream path.

16 Figure 7 illustrates transferring of requests from a second USP request  
17 queue 506(B) to a corresponding USP dispatch queue 508(B). Figure 7 is divided  
18 into three horizontal sections, as indicated by the circled numbers, 4-6. When  
19 processing of USP request queue 506(B) begins, USP request queue 506(B)  
20 contains nine requests (i.e., B1, B2, B3, B4, B5, B6, B7, B8, and B9). The  
21 available upstream bandwidth is represented by box 702 while the available  
22 downstream bandwidth is represented by box 604. Bandwidth management  
23 system 504 determines whether or not the first queued request B1 can be handled  
24 by the available downstream bandwidth 604. (This is represented in Figure 7 by  
25 the dashed box B1 within the downstream bandwidth box 604.) Because upstream

1 paths A and B are associated with the same downstream path, the downstream  
2 bandwidth must be able to accommodate any previously scheduled requests (A1  
3 and A2) as well as any requests to be scheduled. Bandwidth management  
4 system 504 also determines whether or not an expected size of a response to  
5 request B1 can be handled by the available upstream bandwidth 702. (This is  
6 represented in Figure 7 by the dashed box B1 Rsp within the upstream bandwidth  
7 box 702.) If the request can be handled by the downstream bandwidth and the  
8 anticipated response to the request can be handled by the upstream bandwidth,  
9 then the request is transferred from the USP request queue 506(B) to the  
10 corresponding USP dispatch queue 508(B).

11 In the illustrated example, similar to the scheduling of requests from USP  
12 request queue 506(A), the available upstream bandwidth can accommodate two of  
13 the queued requests, but not the third. Therefore, requests B1 and B2 are  
14 transferred to the USP dispatch queue 506(B). Bandwidth management  
15 system 504 continues processing in this manner, one USP request queue at a time,  
16 scheduling as many requests as possible. Also, although, as illustrated, dispatch  
17 queue 508(B) contains requests B1 and B2 at the same time, request B1 may be  
18 dispatched before request B2 is added to dispatch queue 508(B).

19 Figure 8 illustrates processing to schedule requests from USP request  
20 queue 506(D). In the illustrated example, it is assumed that two requests (A1 and  
21 A2) have been scheduled from USP request queue 506(A), two requests (B1 and  
22 B2) have been scheduled from USP request queue 506(B), and three requests (C1,  
23 C2, and C3) have been scheduled from another USP request queue (not shown).

24 When processing of USP request queue 506(D) begins, USP request  
25 queue 506(D) contains nine requests (i.e., D1, D2, D3, D4, D5, D6, D7, D8, and



1 D9). The available upstream bandwidth is represented by box 802 while the  
2 available downstream bandwidth is represented by box 604. Bandwidth  
3 management system 504 determines whether or not an expected size of a response  
4 to request D1 can be handled by the available upstream bandwidth 802. (This is  
5 represented in Figure 8 by the dashed box D1 Rsp within the upstream bandwidth  
6 box 802.) Bandwidth management system 504 also determines whether or not the  
7 first queued request D1 can be handled by the available downstream  
8 bandwidth 604. (This is represented in Figure 8 by the dashed box D1 within the  
9 downstream bandwidth box 604.) Because upstream paths A, B, C, and D are  
10 associated with the same downstream path, the downstream bandwidth must be  
11 able to accommodate any previously scheduled requests (A1, A2, B1, B2, C1, C2,  
12 and C3) as well as any requests to be scheduled. In the illustrated example, the  
13 available downstream bandwidth 604 cannot accommodate queued request D1,  
14 even though the available upstream bandwidth 802 can accommodate the  
15 anticipated response to queued request D1. Accordingly, request D1 is not  
16 transferred from USP request queue 506(D) to USP dispatch queue 508(D).  
17 Furthermore, because the available downstream bandwidth is essentially full,  
18 bandwidth management system 504 does not attempt to schedule any additional  
19 requests for the particular downstream path during the current scheduling interval.  
20 Scheduling does continue, however, for USP request queues associated with other  
21 downstream network paths. In an exemplary implementation, to ensure that  
22 requests in each USP request queue get scheduled, with each new scheduling  
23 interval, bandwidth management system 504 begins scheduling requests from the  
24 first USP request queue from which no requests were scheduled during the  
25 previous scheduling interval. Accordingly, in the example described, USP request

1 queue 506(D) would be processed first in association with the particular  
2 downstream path during the next scheduling interval.

3 In the example described above with reference to Figures 6-8, each USP  
4 request queue 506 had only a single associated set-top box. However, in reality, a  
5 particular upstream network path may support multiple set-top boxes, and  
6 therefore each USP request queue 506 may contain requests directed to any  
7 combination of those multiple set-top boxes. In the above-described scenario,  
8 while data is being sent to the set-top box over the downstream path, the  
9 corresponding upstream path is idle. Similarly, when the set-top box sends a  
10 response to the request, the upstream path is busy, but the downstream path is then  
11 idle. Accordingly, in an exemplary implementation, to reduce network idle time,  
12 and thereby increase performance, communication with multiple set-top boxes that  
13 share the same upstream path may be allowed during a particular scheduling  
14 interval. However, the more set-top boxes that share the same upstream path that  
15 are trying to communicate with the headend at the same time, the higher the  
16 chance of data collisions and lost data packets. Therefore, IAG 208 may also  
17 maintain a configurable **maxConnections** parameter that identifies a maximum  
18 number of set-top boxes that share a single upstream path and can be scheduled to  
19 receive requests during a single request scheduling interval. The smaller the value  
20 of **maxConnections**, the lower the risk of losing data packets.

21 A dispatcher thread associated with a particular dispatch queue controls the  
22 dispatch of scheduled requests based on the **maxConnections** parameter. As an  
23 example, referring back to Figure 6, if **maxConnections** = 1, then requests A1 and  
24 A2 can be dispatched simultaneously if they are both directed to the same set-top  
25 box. On the other hand, if **maxConnections** > 1, then requests A1 and A2 may be

1 dispatched simultaneously even if they are directed to different set-top boxes that  
2 share the same upstream path.

### 3 Alternate Exemplary Scheduling Algorithm

4 When multiple applications 206 are trying to communicate with set-top  
5 boxes over the same downstream and upstream paths, it is important to ensure that  
6 bandwidth is available to each of the applications. Furthermore, some applications  
7 may require more bandwidth than others, more specifically, one application may  
8 require a large amount of downstream bandwidth but only a small amount of  
9 upstream bandwidth, or visa versa. Figure 9 illustrates selected components of an  
10 exemplary interactive application gateway 208 configured to manage bandwidth  
11 for multiple channels of a two-way communication network. Similar to Figure 5,  
12 the IAG 208 shown in Figure 9 includes bandwidth management system 504.  
13 Bandwidth management system 504 includes upstream path (USP) request  
14 queues 506(1), 506(2), 506(3), ..., and 506(Y). Bandwidth management  
15 system 504 further includes corresponding upstream path (USP) dispatch  
16 queues 508(1), 508(2), 508(3), ..., and 508(Y). As described above with-respect  
17 to Figure 5, each USP request queue 506 (and each corresponding USP dispatch  
18 queue 508) corresponds to an upstream path of the out-of-band network. Where  
19 the IAG 208 illustrated in Figure 5 includes an application request queue 502, the  
20 IAG 208 illustrated in Figure 9 includes application request channels 902(1),  
21 902(2), ..., 902(Z). Each application request channel may be configured to use up  
22 to a particular portion of the associated upstream and/or downstream network path.  
23 For example, application request channel 902(1) may be configured to use up to  
24 25% of the downstream path and 75% of the upstream path, and application  
25 request channel 902(2) may be configured to use up to 60% of the downstream

1 path but only 10% of the upstream path. One or more applications 206 may then  
2 be assigned to a particular application request channel 902, indicating that that  
3 requests from that particular application 206 may have access to the designated  
4 upstream and/or downstream bandwidth. Application request channels 902  
5 receive, from one or more applications 206, requests that are to be transmitted to  
6 one or more set-top boxes 108 over the out-of-band network 203. Requests from  
7 applications 206 are transferred from application request channels 902 to  
8 bandwidth management system 504 where the requests are distributed to the  
9 appropriate USP request queues 506 based on which set-top box each request is  
10 directed to. For example, if USP request queue 506(1) corresponds to upstream  
11 path 406 (shown in Figure 4), then any requests directed to set-top boxes 314(1)  
12 or 314(2) are placed in USP request queue 506(1). Similarly, if USP request  
13 queue 506(2) corresponds to upstream path 408, then any requests directed to set-  
14 top boxes 316(1), 316(2), or 316(3) are placed in USP request queue 506(2).

15 Bandwidth management system 504 applies a scheduling algorithm at  
16 regular intervals to transfer requests from USP request queues 506 to the  
17 corresponding USP dispatch queues 508. The requests are then transferred from  
18 the USP dispatch queues 508 to network controller 210, and then on to out-of band  
19 modulator 204 for transmission across out-of-band network 203 to the specified  
20 set-top boxes.

21 Figure 10 illustrates transferring of requests from two USP request  
22 queues 506(A) and 506(B) to corresponding USP dispatch queues 508(A)  
23 and 508(B). At the beginning of the scheduling interval, USP request  
24 queue 506(A) contains nine requests (i.e., A1, A2, A3, A4, A5, A6, A7, A8, and  
25 A9) and USP request queue 506(B) contains seven requests (i.e., B1, B2, B3, B4,

1 B5, B6, B7). In the illustrated example, request A1 was received from an  
2 application request channel 902 that is configured to access up to 25% of the  
3 available downstream bandwidth. Similarly, requests A2 and B1 were received  
4 from an application request channel 902 configured to access up to 15% of the  
5 available downstream bandwidth, and request B2 was received from an  
6 application request channel 902 configured to access up to 10% of the available  
7 downstream bandwidth. (In the illustrated example, the application request  
8 channels do not have specified upstream bandwidth limitations.) The available  
9 upstream bandwidth associated with USP dispatch queue 508(A) is represented by  
10 box 1002, the available upstream bandwidth associated with USP dispatch  
11 queue 508(B) is represented by box 1004, and the available downstream  
12 bandwidth associated with both USP dispatch queues is represented by box 1006.  
13 The downstream bandwidth 1006 is divided up into four channels 1008, 1010,  
14 1012, and 1014, each of which may be associated with a particular application  
15 request channel 902. For example, downstream channel 1008, which includes  
16 25% of the total available bandwidth may be assigned to a first application  
17 channel, downstream channel 1010, which includes 10% of the total available  
18 downstream bandwidth may be assigned to a second application channel 902, and  
19 downstream channel 1012, which includes 15% of the total available downstream  
20 bandwidth may be assigned to a third application request channel 902.  
21 Downstream channel 1014, which includes 50% of the total available downstream  
22 bandwidth may be assigned to a fourth application request channel, or may not be  
23 assigned to an application request channel, but may be accessible to applications  
24 that are not assigned to an application request channel.  
25

1 When scheduling queued requests, bandwidth management system 504  
2 determines whether or not an expected size of a response to request A1 can be  
3 handled by the available upstream bandwidth 1002. (This is represented in  
4 Figure 10 by the dashed box A1 Rsp within the upstream bandwidth box 1002.)  
5 Bandwidth management system 504 also determines whether or not the first  
6 queued request A1 can be handled by the portion of the available downstream  
7 bandwidth assigned to the application request channel through which the request  
8 was received. For example, in the illustrated example, request A1 was received  
9 through an application request channel that is associated with 25% of the total  
10 available downstream bandwidth. Accordingly, bandwidth management  
11 system 504 determines whether or not request A1 can be accommodated with 25%  
12 of the total available downstream bandwidth. (This is represented in Figure 10 by  
13 the dashed box A1 within downstream channel 1008.) Because the request can be  
14 handled by the downstream channel and the anticipated response to the request can  
15 be handled by the upstream bandwidth, the request is transferred from the USP  
16 request queue 506(A) to the corresponding USP dispatch queue 508(A).

17 To schedule request A2, bandwidth management system 504 verifies that an  
18 anticipated response to request A2 can be handled by the available upstream  
19 bandwidth 1002 and that the response can be handled by the downstream  
20 channel 1012 that is associated with application request channel through which  
21 request A2 was received. Requests B1 and B2 are scheduled in a similar way. To  
22 schedule request B1, bandwidth management system 504 verifies that an  
23 anticipated response to the request can be handled by the available upstream  
24 bandwidth 1004 and that the response can be handled by the downstream  
25 channel 1012 that is associated with the application request channel through which

1 request B1 was received. Because requests A2 and B1 were received through the  
2 same application request channel, in order for both requests to be scheduled, both  
3 requests must fit within the downstream channel 1012 associated with the  
4 application request channel through which the requests were received.

5 In an exemplary implementation, less than 100% of the total available  
6 bandwidth (upstream and/or downstream) may be distributed across application  
7 requests channels 902. For example, in a system that supports video-on-demand  
8 (VOD), one or more of the set-top boxes may be allowed to initiate  
9 communication over the out-of-band network, for example, to request a particular  
10 video-on-demand. To reduce the chance of data collisions caused by too much  
11 data traffic and not enough bandwidth to support it, a portion of the total available  
12 upstream bandwidth may be reserved (not assigned to any of the application  
13 request channels) to ensure that at least a portion of the upstream bandwidth is  
14 available when a set-top box initiates a request.

15 Furthermore, in addition to assigning different portions of available  
16 upstream and/or downstream bandwidth to application request channels 902, the  
17 portions that are assigned may vary over time. For example, VOD requests may  
18 be minimal Monday – Friday between the hours of 3:00 am and 8:00 pm, but the  
19 number of VOD requests may increase on the weekends and Monday – Friday  
20 between the hours of 8:00 pm and 3:00 am. Accordingly, bandwidth management  
21 system 504 may be configured to allocate, for example, 100% of the available  
22 upstream bandwidth across the application request channels Monday – Friday  
23 between 3:00 am and 8:00 pm, but to allocate, for example, only 75% of the  
24 available upstream bandwidth across the application request channels on the  
25 weekends and Monday – Friday between 8:00 pm and 3:00 am, thereby reserving

1 25% of the total available upstream bandwidth for use by set-top boxes initiating  
2 requests.

3 For example, if a bandwidth management system is set up to include four  
4 application request channels, each being assigned 25% of the available upstream  
5 bandwidth. If the total upstream bandwidth is 10000 bits in the current scheduling  
6 interval, and 100% of the total bandwidth is available to the application request  
7 channels, then each application request channel can use up to 2500 bits. If,  
8 however, at a certain time of day, 25% of the total upstream bandwidth is reserved  
9 to facilitate set-top box initiated requests, then each application request channel  
10 may only use up to 1875 bits (i.e., 25% of 75% of 10000).

11 Additionally, the portions of bandwidth assigned to particular application  
12 request channels may vary over time. For example, a particular application may  
13 require more bandwidth during a certain time of the day. Accordingly, the  
14 application request channel that the application is associated with may be  
15 assigned, for example, only 10% of the available upstream and/or downstream  
16 bandwidth most of the time, but during the time when the application demands  
17 more bandwidth, the application request channel may be assigned, for example,  
18 50% of the available bandwidth.

### 19 **Handling Failures**

20 In the described exemplary implementation, assuming the maxConnections  
21 parameter equals one, requests are dispatched from a dispatch queue one at a time.  
22 The next request is not dispatched until a response to the previous request is  
23 received or until the previous request times out.

24 Figure 11 illustrates negative effects of repeated request failures.  
25 Boxes 1102 represent a USP dispatch queue associated with a particular upstream



1 network path. In the illustrated example, it is assumed that requests are scheduled  
2 for dispatch every second and that a timeout interval is set to two seconds. It is  
3 also assumed that five requests can be scheduled during any given scheduling  
4 interval. Accordingly, as illustrated in Figure 11, at time  $t = 1$  second, five  
5 requests (i.e., R1, R2, R3, R4, and R5) are added to USP dispatch queue 1102. At  
6 the same time, the first of the scheduled requests (R1) is dispatched.

7 Assuming that the set-top box to which request R1 was dispatched is  
8 unresponsive, request R1 remains pending for two seconds (based on the timeout  
9 period). Because the scheduling interval is only one second, at time  $t = 2$  seconds,  
10 even though no more requests have been dispatched from the queue, five more  
11 requests (R6 – R10) are added to the queue.

12 At time  $t = 3$  seconds, request R1 times out, so request R2 can be  
13 dispatched. In addition, five more requests (R11 – R15) are scheduled.

14 As is apparent from Figure 11, when one or more set-top boxes associated  
15 with a particular USP dispatch queue are unresponsive, the system can get  
16 overloaded with requests that can not be dispatched as fast as they are being  
17 scheduled.

18 Figure 12 illustrates an exemplary USP dispatch queue over time in a  
19 system that handles failures. Boxes 1202 represent a USP dispatch queue over  
20 time. As in Figure 11, it is assumed that requests are scheduled for dispatch every  
21 second and that a timeout interval is set to two seconds. It is also assumed that  
22 five requests can be scheduled during any given scheduling interval. Accordingly,  
23 as illustrated in Figure 12, at time  $t = 1$  second, five requests (i.e., R1, R2, R3, R4,  
24 and R5) are added to USP dispatch queue 1202. At the same time, the first of the  
25 scheduled requests (R1) is dispatched.

1 Assuming that the set-top box to which request R1 was dispatched is  
2 unresponsive, request R1 remains pending for two seconds (based on the timeout  
3 period). Because the scheduling interval is only one second, at time  $t = 2$  seconds,  
4 even though no more requests have been dispatched from the queue, five more  
5 requests (R6 – R10) are added to the queue.

6 To prevent the number of pending requests in USP dispatch queue 1202  
7 from getting too large, a maximum number of pending requests is enforced. In an  
8 exemplary implementation, the maximum number of pending requests is an  
9 integer multiple of the number of requests that can be scheduled during a  
10 particular scheduling interval. In the illustrated example, it is assumed that the  
11 limit is two times the number of requests that can be scheduled, or 10 pending  
12 requests. When the maximum number is reached, only one additional request is  
13 allowed to be scheduled during each successive timeout period. For example, in  
14 the illustrated example, the scheduling interval is every second, but the timeout  
15 period is two seconds. Accordingly, when the maximum number of pending  
16 requests is reached, from that point on, only one request can be scheduled every  
17 two seconds, which is the equivalent of every other scheduling interval.

18 This concept is illustrated in Figure 12 where, at time  $t = 3$  seconds, request  
19 R1 times out, so request R2 can be dispatched.

20 At time  $t = 4$  seconds, request R2 is still pending, but one more request is  
21 scheduled, again bringing the queue to its maximum capacity.

22 In the described exemplary implementation, the rate at which requests are  
23 scheduled increases as set-top boxes become more responsive and more requests  
24 are able to be dispatched during a particular interval.

## **Methods for Reducing Network Contention**

Figure 13 illustrates an exemplary method for scheduling transmission of requests, resulting in reduced network contention. The process shown in Figure 13 is illustrated as a set of operations shown as discrete blocks. The process may be implemented in any suitable hardware, software, firmware, or combination thereof. The order in which the operations are described is not to be construed as a limitation.

At block 1302, IAG 208 receives a request from an application 206. For example, the received request is added to application request queue 502.

At block 1304, IAG 208 identifies the network paths associated with the request. A downstream network path is the path over which the request will be transmitted. An upstream network path is the path over which a response to the request may be received. For example, in an exemplary implementation, each request includes an upstream path ID, a downstream path ID, and an IP address associated with the set-top box to which the request is directed.

At block 1306, IAG 208 identifies other set-top boxes that share the same upstream path ID. The group of set-top boxes that share a particular upstream path are known as contending set-top boxes.

At block 1308, IAG 208 determines whether there are any pending requests that have been transmitted to any of the contending set-top boxes.

At block 1310, when it is determined that there are no pending requests to contending set-top boxes, IAG 208 sends the request, for example, by pushing the request to network controller 210, which then causes the request to be transmitted to the client device.

1 If, at block 1308, it is determined that there is a pending request to a  
2 contending set-top box, then processing continues at block 1308 until there is no  
3 longer a pending request.

4 Figure 14 illustrates an exemplary method for scheduling transmission of  
5 requests based on available bandwidth. The process shown in Figure 14 is  
6 illustrated as a set of operations shown as discrete blocks. The process may be  
7 implemented in any suitable hardware, software, firmware, or combination  
8 thereof. The order in which the operations are described is not to be construed as a  
9 limitation.

10 At block 1402, IAG 208 receives a request from an application 206. For  
11 example, the received request is added to application request queue 502 or the  
12 request is received through an application request channel 902.

13 At block 1404, IAG 208 identifies the network paths associated with the  
14 request. A downstream network path is the path over which the request will be  
15 transmitted. An upstream network path is the path over which a response to the  
16 request may be received. For example, in an exemplary implementation, each  
17 request includes an upstream path ID, a downstream path ID, and an IP address  
18 associated with the set-top box to which the request is directed. In an exemplary  
19 implementation, in addition to identifying the network paths, IAG 208 may group  
20 received requests based on the identified upstream network path, for example, in a  
21 set of upstream path request queues 506.

22 At block 1406, IAG determines whether or not there is enough available  
23 downstream bandwidth to transmit the request. In an exemplary implementation,  
24 multiple requests may be scheduled for transmission during a particular time  
25 interval, and the available downstream bandwidth must be able to accommodate

1 all of the requests that are scheduled from transmission across the particular  
2 downstream path during the particular time interval.

3 In an implementation in which requests are received through application  
4 request channels, each channel may have associated bandwidth allocations. For  
5 example, requests received through a particular channel may only be allowed to  
6 use up to a certain percentage of the available downstream bandwidth. In this  
7 case, in addition to verifying that there is enough downstream bandwidth to  
8 support all of the scheduled request transmissions, the IAG also verifies that the  
9 total bandwidth to be used for requests received through a particular application  
10 request channel does not exceed the specified downstream bandwidth allocation  
11 associated with the application request channel.

12 At block 1408, IAG determines whether or not there is enough available  
13 upstream bandwidth to accommodate an anticipated response to the request. In an  
14 exemplary implementation, each time a response is received, IAG adjusts a  
15 running average response size. This value is then used to determine the required  
16 upstream bandwidth. If the system is configured to allow pending requests to  
17 multiple set-top boxes that contend for the same upstream network path, the  
18 available upstream bandwidth must be able to accommodate anticipated responses  
19 to all of the pending requests to the contending set-top boxes.

20 As described above, in an implementation in which requests are received  
21 through application request channels, each channel may have associated  
22 bandwidth allocations. For example, responses to requests received through a  
23 particular channel may only be allowed to use up to a certain percentage of the  
24 available upstream bandwidth. In this case, in addition to verifying that there is  
25 enough upstream bandwidth to support all of the anticipated responses to

1 contending pending requests, the IAG also verifies that the total upstream  
2 bandwidth to be used for responses to requests received through a particular  
3 application request channel does not exceed the specified upstream bandwidth  
4 allocation associated with the application request channel.

5 At block 1410, when it is determined that there is enough downstream  
6 bandwidth to support the request and that there is enough upstream bandwidth to  
7 support an anticipated response to the request, IAG 208 schedules the request for  
8 transmission. For example, in the above-described exemplary implementation, the  
9 request is transferred from an upstream path request queue 506 to a corresponding  
10 upstream path dispatch queue 508.

11 At block 1412, after the request is scheduled or if it is determined that there  
12 is not enough downstream bandwidth to support the request (the “No” branch from  
13 block 1406) or if it is determined that there is not enough upstream bandwidth to  
14 support an anticipated response to the request (the “No” branch from block 1408),  
15 IAG 208 waits for the next scheduling interval before beginning the process again.

16 Figure 14 illustrates scheduling one request per scheduling interval, merely  
17 for ease of explanation. However, it is recognized that multiple requests may be  
18 scheduled during any given scheduling interval, depending on how the IAG 208 is  
19 configured.

#### 20 **Method for Estimating Request Response Size**

21 Figure 15 illustrates an exemplary method for gathering data to be used to  
22 reduce network contention when scheduling transmission of requests. The process  
23 shown in Figure 15 is illustrated as a set of operations shown as discrete blocks.  
24 The process may be implemented in any suitable hardware, software, firmware, or  
25

1 combination thereof. The order in which the operations are described is not to be  
2 construed as a limitation.

3 At block 1502, IAG 208 receives a request from an application.

4 At block 1504, IAG 208 schedules transmission of the request. Example  
5 methods for scheduling request transmission are described above.

6 At block 1506, IAG 208 transmits the request (or, as described above,  
7 transfers the request to a network controller 210, which then causes the request to  
8 be transmitted).

9 At block 1508, IAG 208 receives a response to the request.

10 At block 1510, IAG 210 calculates an average response size. In an  
11 exemplary implementation, a running average response size is maintained, being  
12 re-calculated each time a response is received.

13 At block 1512, IAG 210 forwards the received response to the application  
14 from which the corresponding request was received.

### 15 16 **Method for Handling Request Failures**

17 Figure 16 illustrates an exemplary method for handling request failures,  
18 such as timeouts, so as to not schedule an excessive number of pending requests.  
19 The process shown in Figure 16 is illustrated as a set of operations shown as  
20 discrete blocks. The process may be implemented in any suitable hardware,  
21 software, firmware, or combination thereof. The order in which the operations are  
22 described is not to be construed as a limitation.

23 In an exemplary implementation, the process illustrated in Figure 16 is  
24 performed each time the IAG 208 prepares to transfer a request from a USP  
25 request queue to a corresponding USP dispatch queue.

1 At block 1602, IAG 208 determines the number of pending requests in the  
2 particular USP dispatch queue.

3 At block 1604, IAG 208 compares the number of queued requests to a  
4 maximum number of queued requests. In an exemplary implementation, the  
5 maximum number is configurable to be, for example, an integer multiple of the  
6 number of requests that can be scheduled into the dispatch queue during a  
7 scheduling interval.

8 At block 1606, when it is determined that the number of pending requests  
9 that are queued is less than a maximum number of allowed pending requests (the  
10 “No” branch from block 1604), the request is scheduled as described in more  
11 detail above with reference to earlier figures..

12 At block 1608, when it is determined that the maximum number of pending  
13 requests are queued (the “Yes” branch from block 1604) or after a request has  
14 been scheduled, IAG 208 waits for the next scheduling interval, at which time  
15 processing continues as described above with reference to block 1602.

## 16 17 **Conclusion**

18 As described above, shared network path contention can be reduced by  
19 identifying client devices that share downstream and/or upstream network paths  
20 and managing both the downstream bandwidth across which requests are send and  
21 the upstream bandwidth across with responses are received. Although the systems  
22 and methods have been described in language specific to structural features and/or  
23 methodological steps, it is to be understood that the invention defined in the  
24 appended claims is not necessarily limited to the specific features or steps  
25



1 described. Rather, the specific features and steps are disclosed as preferred forms  
2 of implementing the claimed invention.  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25